NPS-OR-03-001

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Applicability Assessment for Maritime Operations Simulation (MarOpsSim) – Phase II

by

Arnold H. Buss

Thomas E. Halwachs

February 2003

Prepared for: U.S. Coast Guard Research and Development Center
Maritime Operations Technology Division
1082 Shennecossett Rd.
Groton, CT 06340

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5000

RADM David R. Ellison
Superintendent

Richard Elster
Provost

This report was prepared for and funded by the United States Coast Guard Research and development Center, Maritime Operations Technology Division, 1082 Shennecossett Rd., Groton, CT 06340.

This report was prepared by:

ARNOLD H. BUSS
Research Professor of
Operations Research

THOMAS E. HALWACHS
Senior Lecturer of Operations Research

Reviewed by:

R. KEVIN WOOD
Associate Chairman for Research
Department of Operations Research

JAMES N. EAGLE
Chairman
Department of Operations Research

Released by:

DAVID W. NETZER
Associate Provost and Dean of Research

REPORT DOCUMENTATION PAGE			Form approved OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 2003	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE Applicability Assessment for Maritime Operations Simulation (MarOpsSim) – Phase II	5. FUNDING DTCG32-01-X-R00
6. AUTHOR(S) Arnold H. Buss and Thomas E. Halwachs	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER NPS-OR-03-001
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) United States Coast Guard Research and Development Center Maritime Operations Technology Division 1082 Shennecossett Rd. Groton, CT 06340	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

The Maritime Operations Simulation (MarOpsSim) is a modern discrete event simulation being used by the U.S. Coast Guard. MarOpsSim was originally developed by the U.S. Coast Guard Research & Development Center to examine elements of the Search & Rescue and Law Enforcement missions. MarOpsSim is now being extended to encompass all 14 Deepwater missions to support the Deepwater Acquisition. This document reports the results of the Core Verification and Validation (Core V&V) effort conducted at the Naval Postgraduate School by the authors. Based on the analysis and assessment made so far, it can be concluded that MarOpsSim is a modeling tool that can be used to reasonably represent the characteristics and behavior of the Coast Guard against the demands of, and in the operating environments described in, the Modeling and Simulation Master Plan (MSMP). Furthermore, MarOpsSim produces output that can be summarized and analyzed to consistently generate Coast Guard system performance measures also described in the MSMP.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Discrete-Event Simulation; United States Coast Guard; Deepwater Acquisition; Research & Development Center	27
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unlimited	Unlimited	Unclassified	Unlimited

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18

Executive Summary

This document represents Phase II of the Verification and Validation (V&V) effort for the Maritime Operations Simulation Model (MarOpsSim) under development for the United States Coast Guard (USCG). Phase I of this effort documented the validation of MarOpsSim's core features.

The Phase II effort is called an "Applicability Assessment" and represents a high-level assessment of MarOpsSim as a modeling tool that can reasonably represent the characteristics and behavior of Coast Guard demands, assets, and operating environments as described in the Coast Guard's Modeling & Simulation Master Plan (MSMP). The Phase II report also provides general guidance for follow-on V&V activities and identifies specific areas where modelers should pay particularly close attention when validating MarOpsSim applications. Lastly, areas where the MarOpsSim's core has changed since Phase I are re-examined in the Phase II study.

Based on the Phase I analysis of the core MarOpsSim components and the review of the relevant Deepwater documents, a number of conclusions can be made regarding the applicability of MarOpsSim to the Deepwater Acquisition Process. MarOpsSim does model the four essential components of the proposed Integrated Deepwater System: Surface Platforms, Air Platforms, Logistics, and C4ISR. MarOpsSim provides, either through core capabilities or through a flexible and expressive scripting language, the ability to model each of these overlapping system elements to support proposed Integrated Deepwater System features, and thereby support the Deepwater selection process.

Based on the analysis and assessment made so far, it can be concluded that MarOpsSim is a modeling tool that can be used to represent the characteristics and behavior of the Coast Guard against the demands and in the operating environments described in the MSMP. Furthermore, MarOpsSim produces output that can be summarized and analyzed to consistently generate the Coast Guard system performance measures also described in the MSMP.

Table of Contents

List of Figures

List of Tables

1 Introduction

This document represents Phase II of the Verification and Validation (V&V) effort for the Maritime Operations Simulation Model (MarOpsSim) under development for the United States Coast Guard (USCG). Phase I of this effort documented the validation of MarOpsSim's core features.

1.1 Scope & Objectives

The Phase II effort is called an "Applicability Assessment" and represents a high-level assessment of MarOpsSim with respect to its use as a modeling tool that can reasonably represent the characteristics and behavior of Coast Guard demands, assets, and operating environments described in the USCG Modeling & Simulation Master Plan (MSMP). In addition, the Phase II report provides general guidance for follow-on V&V activities and identifies specific areas where modelers should pay particularly close attention when testing MarOpsSim behavior. Since Phase I, the model has undergone some changes. In those areas, the Phase II study briefly revisits the Phase I work and identifies elements of the MarOpsSim core that may require revalidation.

To that end, the following steps were taken:

- Reviewed Modeling and Simulation Master Plan (MSMP) through Change 9.
- Reviewed the Consolidated Software Design Document (CSDD) dated 15 December 2000.
- Reviewed the Modeled Concept of Operations for Coast Guard Legacy Assets dated 15 June 2001.
- Attended Operational Effectiveness Workshops with USCG representatives and the MarOpsSim contractor.
- Reviewed Phase I V&V effort in light of changes to MarOpsSim that had been made subsequent to Phase I.

1.2 Document Overview

The report documents a review and assessment of MarOpsSim with respect to its use as a modeling tool that can reasonably represent the characteristics and tactics of Coast Guard demands, assets, and operating environments described in the MSMP. The document is organized into the following sections to reflect the three phases of this task. An overview of their contents is as follows:

- **Section 1: Introduction** – Provides the background for the project, a statement of the project objectives, and an overview of the document.
- **Section 2: Applicability Assessment** – Addresses the applicability of the Deepwater MarOpsSim Application developed using MarOpsSim to the goals of the Deepwater Acquisition Program.
- **Section 3: Validating the Deepwater MarOpsSim Application** – Provides general guidance for validating applications built to use MarOpsSim, focusing on application-specific scripted elements for Communications, Tactics, Command and Control, and Operational Picture.
- **Section 4: Review of Core Validation** – Briefly reviews Phase I and documents the revalidation of MarOpsSim algorithms that have been modified since Phase I.

2 Applicability Assessment

This section addresses the applicability of the Deepwater MarOpsSim Application developed using MarOpsSim to the goals of the Deepwater Acquisition Program. Specifically:

- Is MarOpsSim the appropriate tool for modeling the MSMP demands and environment?
- Is the Deepwater MarOpsSim Application at the correct level of fidelity to produce meaningful results?

These questions are addressed through examination of the fidelity of the Deepwater MarOpsSim Application and the comparison of the scope of the Deepwater MarOpsSim Application to the Deepwater Program.

2.1 Deepwater MarOpsSim Application Fidelity

There exist two issues concerning the fidelity of the Deepwater MarOpsSim Application as it applies to MarOpsSim and the goals of the Deepwater Program.

1. Is the fidelity of the Deepwater MarOpsSim Application appropriate to be modeled using MarOpsSim?
2. Is the fidelity of the Deepwater MarOpsSim Application appropriate to support Deepwater assessments?

The Deepwater MarOpsSim Application is a campaign-level model consisting of platform-level entities. It consists of multiple scenarios modeling Coast Guard operations regionally over a period of one year. Each Coast Guard asset, both aircraft and surface ship, is modeled with its own characteristics, capabilities, and sensors. This large-scale effort is broken down into four regions and four levels of demand.

A campaign-level model with platform-level entities created in MarOpsSim of a regional scenario will be a medium-to-large-scale model. Phase I of MarOpsSim IV&V has demonstrated that MarOpsSim is capable of executing such models, so studies of this kind are feasible. Therefore, MarOpsSim is a practical tool for creating and analyzing these scenarios.

The use of platform-level entities in a campaign-level model is appropriate for Coast Guard use, especially in the large-scale scenario. While run times may be long, the value of determining meaningful estimates of the Performance Indicators would enable the Coast Guard to evaluate various force structures against different demands.

The performance measures outlined in the MSMP are in the form of Operational Effectiveness Indicators (OEIs). Some key points of analysis involve comparing the performance of various assets, based on their modeled capabilities. In order to adequately capture the characteristics of these systems and evaluate their impact on the suite of OEIs, it is best to explicitly model the platforms themselves, as the Deepwater MarOpsSim Application does. MarOpsSim provides the capabilities to model these platforms, as discussed previously. Modeling assets at a less aggregate level would increase model run times significantly without any appreciable increase in accuracy. Therefore, the Deepwater MarOpsSim Application level of resolution is the best one for the proposed U.S. Coast Guard use.

Individual components of the Deepwater MarOpsSim Application modeled using MarOpsSim are discussed in further detail in the following sections.

2.1.1 Scope of Model

The Deepwater MarOpsSim Application scenarios will model Coast Guard operations for a period of one simulated year against four demand levels in four regions. Model applicability requires appropriate accuracy in calculation of OEIs, while providing a sufficient number of iterations (data points feeding the overall results analysis). The number of replications is a parameter set in the Deepwater MarOpsSim Application configuration.

The region-level scope with a time horizon of a year has a number of implications for the applicability of a model. The level of resolution for the Deepwater MarOpsSim Application must be commensurate with that required for the kind of analysis envisaged by the Deepwater Acquisition Program. Since OEIs are collected at the regional level, the fidelity of the modeled components needs to be at a corresponding level.

The following paragraphs discuss the applicability of MarOpsSim's modeling approach in the areas of Platform Movement, Sensors, Communications, Command and Control, and Tactics. The discussion focuses on whether MarOpsSim's modeling approach is appropriate to the type of analysis conducted.

2.1.2 Platform Movement

The Deepwater MarOpsSim Application utilizes MarOpsSim's basic platform movement model, which is piecewise-constant motion along a series of waypoints. The speed on any segment can be set to a legitimate value between the platform's minimum and maximum speed. Fine-grained motion, such as acceleration and detailed turning, are not modeled. MarOpsSim constrains movement through land-avoidance algorithms. These algorithms serve two purposes: to ensure that sea-based platforms do not maneuver in restricted waters or over land, and to ensure that air-based platforms do not fly over restricted air space.

The implementation of the platform motion and land-avoidance algorithms was verified in the Phase I Validation study [Buss, A. & T. Halwachs. Core Validation for Maritime Operations Simulation (MarOpsSim), Technical Report NPS-OR-00-093 (November 1999)] and are being reverified as part of the current Phase II study.

MarOpsSim's approach to platform movement is reasonable for the level of detail in scenarios. No appreciable improvement in Performance Indicator estimates would be gained by modeling movement in more detail. For example, the ability of a dispatched cutter to reach a destination over the regional distances in the scenarios would largely be a function of its cruising speed. Modeling acceleration, deceleration, and turning radius does not appreciably affect the OEIs. Furthermore, incorporating detailed motion algorithms would significantly increase model run time without a proportionate increase in the accuracy of the results.

2.1.3 Sensors

The Deepwater MarOpsSim Application uses MarOpsSim's Probability of Detection (POD) vs. Range curve approach to modeling the detection of targets by sensors. This methodology is standard and is used in many analytical studies. For the platform-centric resolution of MarOpsSim, modeling individual detections is the appropriate level of resolution. An alternative, such as one involving the rate at which platforms are detected without invoking individual detections, would be too coarse. Conversely, a highly detailed, engineering-level model of sensors would be far too fine for the regional scenarios modeled in the Deepwater MarOpsSim Application. Such an engineering-level model would waste

precious CPU time and result in outcomes that would have no more accuracy than those using the POD/Range curves implemented in MarOpsSim.

2.1.4 Communications and Command and Control

Communications are modeled in MarOpsSim using messaging events associated with communications hardware assigned to each platform. Implementation of the communications capabilities has significant impact on translating the operational picture throughout the modeled domain, which impacts tactical decisions. The operational picture of each platform must only be developed from its own sensor inputs and received communications. While the Core Validation addressed the functionality message passing in MarOpsSim, it is noted that the application-specific validation discussed in **Section 3: Validating the Deepwater MarOpsSim Application** includes verification of actions based on communications.

This is the appropriate level of resolution for modeling communications in platform-centric models such as MarOpsSim. The communications component is utilized in the Deepwater MarOpsSim Application to model the transfer of command and control, as well as operational picture information. Modeling communications at the message level is critical to the Deepwater MarOpsSim Application because of the effect on command decisions, the development of an operational picture, and the importance of command and control to the Deepwater system. Additional levels of fidelity would add unnecessary complexity without any gains in precision. Therefore, MarOpsSim provides the level of communication that supports the OEIs, as well as the goals of the Deepwater Program.

2.1.5 Tactics

Tactics are implemented in MarOpsSim via scripts that prescribe behaviors to the individual platforms. Scripting at the platform level is an appropriate level of resolution given the level of detail of the other component modeled in the Deepwater MarOpsSim Application. The scripting capability is sufficient to define default behaviors (e.g., a patrol route selection and execution), as well as for defining responses to the environment or operational picture (e.g., the response to detection of a suspicious vessel). The flexibility of the MarOpsSim scripting language enables a modeler to capture the essential impact of tactics and doctrine in the performance indicators estimated by the Deepwater MarOpsSim Application. This approach is the appropriate level of resolution for the uses envisioned for the Deepwater MarOpsSim Application.

2.1.6 Conclusions

There is a misperception by some modelers that more detail in a model necessarily leads to a better or more appropriate model. In fact, too much detail can be counterproductive and possibly lead to less accurate estimates of OEIs. The problem is that every part of the model has errors inherent in the abstraction in creating the model. The more entities that comprise the model, the greater the additive effect of errors. In some cases there can be cancellation of the errors, but that should not be counted on. A useful analogy is the modeling of the temperature in a room. Physically, temperature is an aggregate measure of the collective energy in the air molecules. Therefore, a detailed model would consist of modeling the energy of each air molecule and estimating the performance measure (temperature) as the sum of each molecule's energy. A less detailed model would consist of placing a thermometer in the room and observing its value after a "warm-up" period. Even if the former experiment could be performed, it should be clear that its estimate would be wildly off, whereas the latter, more aggregate, model would be far more accurate.

These issues are relevant to MarOpsSim applications because of the inherent fidelity of MarOpsSim. For example, the MarOpsSim sensor components utilized by the Deepwater MarOpsSim Application are not extremely detailed engineering-level models. Rather, the Deepwater MarOpsSim Application utilizes MarOpsSim algorithms that focus on the times that sensors detect and acquire targets. These algorithms can be calibrated to more detailed data from the sensors, and thereby provide detection capabilities that are at the appropriate level of resolution to the rest of the scenario.

Using a platform-level model like MarOpsSim for region-level analysis is, therefore, quite reasonable. Ideally there should be a close correspondence between the level of detail for the questions being asked and the level of resolution of the models being used to answer them. MarOpsSim is most effective as a modeling framework at a level of detail just below the region level. It is ideally suited for smaller locales over somewhat shorter periods of time. However, as long as care is taken, it is suitable for analyzing an entire region.

It is, therefore, reasonable to use MarOpsSim for the region-level analysis contemplated for Deepwater analysis. While the platform level of detail may involve long run times, the scenarios themselves should be valid for system comparison purposes, provided that they are validated.

2.2 Performance Indicators

There are two issues to be addressed in assessing MarOpsSim's applicability with respect to OEIs. First, are models built using MarOpsSim capable of obtaining the statistics required for estimating the OEIs? This is a matter of determining whether MarOpsSim supports the definition and collection of such values and whether the Deepwater MarOpsSim Application collects them appropriately. The second issue is whether the estimated OEIs can reasonably be used in a meaningful manner; that is, for the purposes intended for Deepwater. Some issues discussed in this section apply to **Section 3: Validating the Deepwater MarOpsSim Application**, but they are included here for purposes of continuity.

The Coast Guard's MSMP lists 66 OEIs in support of the USCG's 14 missions plus the "Operational Picture" [Ref MSMP Appendix E]. Since the applicability of MarOpsSim is tied directly to its ability to estimate these OEIs, it is essential that each OEI be examined closely. It is important to validate the Deepwater MarOpsSim Application's ability to determine and record the information needed to calculate each OEI. Equally important is the calculation of each OEI outside of MarOpsSim – once the simulation is complete.

Some of these OEIs present difficulties for both "real" and simulated systems. Based on the assessment of all the OEIs, one can conclude that the Deepwater MarOpsSim Application is capable of estimating the OEIs within MarOpsSim where there are no external difficulties with the measures themselves. That is, Deepwater MarOpsSim Application is essentially on par with the "real" Coast Guard systems in estimating these measures.

As discussed above, the Deepwater MarOpsSim Application is at least as capable as the actual USCG system in estimating the Performance Indicators that are of interest to the Coast Guard. Therefore, MarOpsSim is a reasonable tool to create and analyze scenarios in support of Deepwater Acquisition. It must be noted that while MarOpsSim is capable of estimating the desired OEIs, the actual implementation is the responsibility of the modelers designing and implementing the scenario.

It is important to recognize that it is quite difficult, if not impossible, for *any* model to produce estimates for OEIs that have absolute accuracy. MarOpsSim is no exception to this rule, and it would be an unfair standard to hold it to. The primary use of MarOpsSim is to provide relative assessments of different proposed Deepwater systems. Therefore, the important issue is to assess the Deepwater MarOpsSim Application's ability to produce consistent *relative* estimates for the different Performance Indicators rather than absolute ones.

A model such as the Deepwater MarOpsSim Application can be used to evaluate systems, in a comparative capacity, that have not yet been implemented or systems about which not enough experimental data are available. Indeed, a role of the simulation model in those cases may be to help guide the design and development of the real system. It should be noted that no specific model can be absolutely validated for all purposes; rather, the most that can be reasonably expected is that a model be validated "for its intended purpose." The role of a modeling tool such as MarOpsSim is to provide the capability to create valid models to support these activities. Thus, the validation issue is whether MarOpsSim provides this capability and whether the MarOpsSim components that are used to create the actual Deepwater MarOpsSim Application scenarios are valid components.

A second issue involves whether a simulation model is to be used for estimating absolute measures of performance as opposed to relative indicators of performance. Use of simulation models for the latter use leads to somewhat less stringent validation requirements, since the model need not necessarily produce valid absolute estimates of performance indicators, but rather be reasonably accurate in estimating the difference (or ratio) of the performance indicators of the two systems. In other words, the user of the simulation model should be confident that if one system has superior performance to another in the simulation model, then the corresponding real system would likewise have superior performance. There are two levels to this relative criterion: At a minimum, the simulation model should correctly identify the order of systems but not necessarily be accurate in estimating the differential between them. An additional level of validity occurs when the simulation model is also accurate in measuring the difference between the systems.

In some cases, data available for scenarios are similar enough to the contemplated scenarios to be used for validation purposes. In these cases, the real data should be used to "drive" the simulation model and the results compared with the actual outcomes. The simulation model is populated with simulated assets that correspond to which assets were actually available for the historical period. The model is then run using the actual inputs and performance indicators collected. Finally, the estimated performance indicators are compared with the actual ones obtained.

If absolute validity is required, then the simulation model should produce measures that are close to the real ones. Ideally, the simulated OEIs should be statistically indistinguishable from the actual values. However, if relative performance is all that is desired, then the simulation model need not necessarily produce estimates that are close to the true ones.

One test that could be performed involves collating output data from the simulation model that correspond to real data collected by the USCG. Personnel who routinely monitor and review the real USCG data would examine the simulated output data for realism and credibility. If they cannot discriminate between real and simulated data after consideration of the simulation context, then it is a good indication that MarOpsSim is producing reasonably valid data. If they can discriminate between the data, the experts should be questioned as to how they were able to distinguish them. This information should be fed back to the developers of that MarOpsSim scenario and appropriate adjustments made.

2.3 Conclusions

Based on the Phase I analysis of the core MarOpsSim components and the review of the relevant Deepwater documents, a number of conclusions can be made regarding the applicability of MarOpsSim to the Deepwater Acquisition Process. MarOpsSim does model the four essential components of the proposed Integrated Deepwater System: Surface Platforms, Air Platforms, Logistics, and C4ISR. MarOpsSim provides, either through core capabilities or through a flexible and expressive scripting language, the ability to model each of these overlapping system elements to support proposed Integrated Deepwater System features and thereby support the Deepwater selection process.

Therefore, based on the analysis and assessment made so far, it can be concluded that MarOpsSim is a modeling tool that can be used to represent the characteristics and tactics of Coast Guard demands, assets, and operating environments described in the MSMP. Furthermore, MarOpsSim produces output that can be summarized and analyzed to consistently generate the Coast Guard system performance measures also described in the MSMP.

As described above, the design and execution of MarOpsSim is consistent with the specifications outlined in the Consolidated Software Design Document. Hence, MarOpsSim models can provide output and data that can be used for relative performance comparisons between proposed systems. In particular, current capabilities of USCG assets, together with current operational tactics and doctrine, can be compared with proposed improved assets, tactics, and doctrine using MarOpsSim. Moreover, the flexibility of MarOpsSim's scripting capabilities make it possible to use it to evaluate any further assets or tactics that could be used by either legacy systems or proposed systems.

3 Validating the Deepwater MarOpsSim Application

Applications are built in MarOpsSim through data input and scenario and tactic scripting. The application validation process must include an internal V&V of the scripts as well as the input data. This section describes the key scripted behaviors that should be included in the V&V effort.

MarOpsSim's scripting capabilities represent a key feature of MarOpsSim, since they fundamentally impact the ability of MarOpsSim to appropriately represent complicated features in a simulation model or new features. MarOpsSim scripts extend the basic functionality of the core by defining new classes, instantiating (allocating) instances of those classes, defining behaviors for classes, and defining general functions. Scripts are also used for configuration and control of scenarios. MarOpsSim is packaged with a library of scripted functions that were included in the Phase I validation. While these functions do not need to be revalidated, the interaction between the library and the application scripts does need to be part of the V&V.

Most scripting features are self-evident in their validity through normal usage and examination of scenarios. For example, a scripting element that configures a simulation run to stop at a particular time can be tested by running the model and ensuring that the simulation halts at the expected time. A similar *pro forma* validation should be done for the set of built-in scripting functions (Appendix C, CSDD).

3.1 Communications

Communications capabilities consist of messages that are sent to platforms that are in range of the sending object. As noted earlier, the scripted behavior dependent upon the transmission or receipt of a message needs to be validated. V&V should confirm that the expected behavior does occur, with no unexpected results. **Section 4: Review of Core Validation** addresses the implementation by the MarOpsSim core.

3.2 Tactics

MarOpsSim's scripting capabilities are used to implement tactics. All tactics that were not tested in the Core Validation should, in principle, be tested with a detailed event audit. For such tactics it is also important to document the precise tactical rules and the platforms affected. For example, the Core Validation study verified that the land avoidance algorithm was correctly implemented by defining the behavior and constraints (surface platforms were confined to water regions) as well as the platforms affected (aircraft were not affected by land masses, except for no-fly zones). Then these behaviors were

evaluated with an event audit that verified the avoidance of land. Thus, the recommended procedure for testing tactics is:

- Define precisely the tactic to be modeled, including constraints.
- Specify the affected platforms.
- Define the regions that are appropriate for these tactics.
- Define the scenarios to be evaluated, specifying the platforms to be tested and the exact tactic that should be observed. The tactics should be expressed in terms of event sequences and state expected behaviors.
- Implement the tactics in scripts and databases.
- Execute the model.
- Verify the sequence of events and state changes specified in Step 4 through a detailed event audit.

Since the MarOpsSim scripting capabilities support the definition of complex behaviors, it is critical that all aspects of the defined behavior are tested. For example, different missions are assigned different priorities and the assets' behavior should reflect those priorities. When an asset is faced with a list of mutually exclusive tasks or missions, it should (presumably) choose the task with the greatest priority, breaking ties with some arbitrary rule such as first-come, first-served.

3.3 Command and Control

MarOpsSim documentation discusses Command and Control (C^2) solely in terms of sensors and communications. It is important that the implemented tactics consistently model realistic United States Coast Guard doctrine. The MarOpsSim scripting environment definitely supports modeling C^2 as defined by sensing/communication, and the Deepwater MarOpsSim Application does exercise this option. Furthermore, MarOpsSim scripts are fully capable of modeling more complex interactions and doctrine.

3.4 Operational Picture

MarOpsSim supports the modeling of individual platform operation pictures by controlling access to information obtained via communications and sensors. **Section 4: Review of Core Validation** addresses the integrity of target information. Like C^2, the use of this information and how it affects the application goals and measures should be verified.

3.5 Validation Conclusions

This assessment concludes that MarOpsSim's scripting capabilities do meet the requirements of the USCG for the Deepwater acquisition process. This means that through scripts it will be possible to construct valid scenarios for purposes of evaluating alternate integrated Deepwater systems. However, this does not in itself guarantee the validity of the particular use of the scripts. As noted above, the Applicability Assessment effort has not attempted to validate all possible uses of MarOpsSim, since that would involve validating particular scenarios populated by particular databases and particular scripts. It is suggested that there be an internal validation process by which certain scripts have been validated for particular uses.

Prevalidating certain scripts would enable their use in a component-like manner for producing scenarios for analysis more rapidly and having more "out-of-the-box" validity. Where possible, these scripts should be documented as to their intended use and calibrated against known test data from actual

systems. Experts in the systems being modeled should be involved with this process to internally validate the applications built using MarOpsSim. An example of the use of experts is similar to the above: Create simple scenarios with the scripts to be tested representing the desired systems, and generate output data in a format similar to those of real data. An inability by the experts to discriminate between actual and simulated output is an indication that the script in question is reasonable.

4 Review of Core Validation

Some core features have been enhanced since the Phase I Core Validation. Since these changes to the core are mostly implementation changes, not design changes, their revalidation was straightforward. The primary components of the core that required revalidation were the Movement, Sensor, and Communication algorithms. Each component was thoroughly tested and is operating as designed. These tests, along with the Phase I tests, revalidate MarOpsSim's core algorithms and basic functions.

4.1 Random Number Generator

The code for the random number generator has not been modified since the Core Validation study. Therefore, revalidation of the random number generator is not necessary.

4.2 Movement Algorithm

The MarOpsSim core has been enhanced to internalize movement algorithms that had previously resided in the script library. The following scenario was used to revalidate the core simulation of platform movement. This scenario is similar to the MonaPass Unrestricted Movement scenario, used in the Phase I validation, but incorporates the change that added the maneuver function to the core.

An asset travels along the points depicted in **Figure 1: Movement Algorithm – Sequence of Points.**

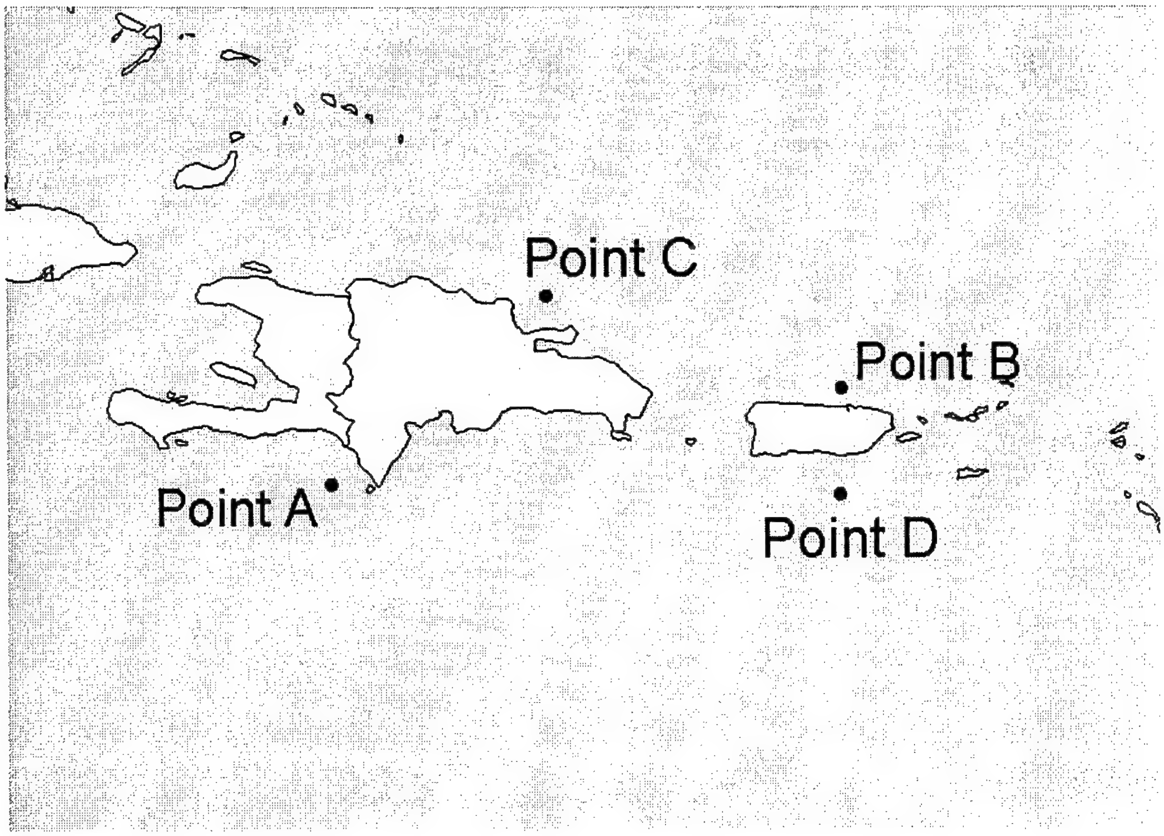

Figure 1: Movement Algorithm – Sequence of Points

The sequence of points used is: A → B → D → C → A → C → D → B → A. Land was included in the figure for reference only and was not present in the scenario. Assets were able to travel directly from point to point with no intermediate points required. This was done so that expected transit times could be calculated based on fixed distances between points. The actual points were located at: Point A (17.6, -71.95); Point B (18.65, -66.19); Point C (19.7, -69.5); and Point D (17.5, -66.2).

The scenario was run for an entire year using two different types of assets moving at two different speeds, as depicted in **Table 1: Asset and Speed Combinations.** In additions, sea state – weather conditions – was randomly changed throughout the year to take into account speed adjusted for surface assets based on sea state.

Table 1: Asset and Speed Combinations

Test	Asset Type	Speed Type
1	WMEC 270	Transit Surveillance
2	WMEC 270	Intercept
3	HH-65	Transit Surveillance
4	HH-65	Intercept

A summary table was created in which platform movement results were compiled from raw results from each of the runs. Descriptions of the columns in the summary table are as follows:

- DestLat – latitude of the destination location

- DestLon – longitude of the destination location
- DestPoint – name of the destination location
- Desired Speed – expected platform speed in miles per hour
- Dist To Point – distance in nautical miles from the current location to the destination location
- Transit Time – time in hours to travel to the destination location based on distance and desired speed
- ETA – estimated time of arrival based on transit time
- Seastate – weather sea state (High, Medium, or Low)
- Speed Type – speed set for the platform
- Arrived Point – location arrived at
- Actual Speed – actual platform speed in miles per hour
- Actual Heading – actual platform heading
- Actual Lat – latitude of the current location
- Actual Lon – longitude of the current location

Each record in this table summarizes what is done at each time step in the model. The estimated time of arrival and destination can be compared with the actual time and destination by looking at adjacent lines in the summary table. This comparison indicates that platform movement is working correctly, with platforms arriving at the time and place expected.

4.3 Land Avoidance Algorithm

The land avoidance algorithms have not undergone modifications and therefore need no further validation.

4.4 Sensor Algorithms

There was only one minor change in the way in which the Sensor Algorithms retrieved data from the POD vs. Range curves. The IV&V of the POD vs. Range is detailed in the next section. In addition, the sensor behaviors "Out of Range" and "Blind/Invisible" were examined and validated.

4.4.1 POD vs. Range

MarOpsSim uses POD vs. Range curves to determine detection events. MarOpsSim's implementation of these algorithms was verified in Phase I, in which it was found to be correctly implemented. Since the completion of Phase I, the data structures for the POD/Range definition data has been refined. These changes warrant a revalidation of MarOpsSim's sensor algorithms. Since the basic theory is identical and the fundamental algorithms have not changed, the validation has been limited to the proper processing of the data in its new format.

The following scenario was used to validate that the core is selecting the correct POD curve based on weather condition, random draw, sensor type, and target size. A target traveling due north is launched every 20 hours at a stationary asset located directly in its path, as depicted in **Figure 2: POD vs. Range Scenario.**

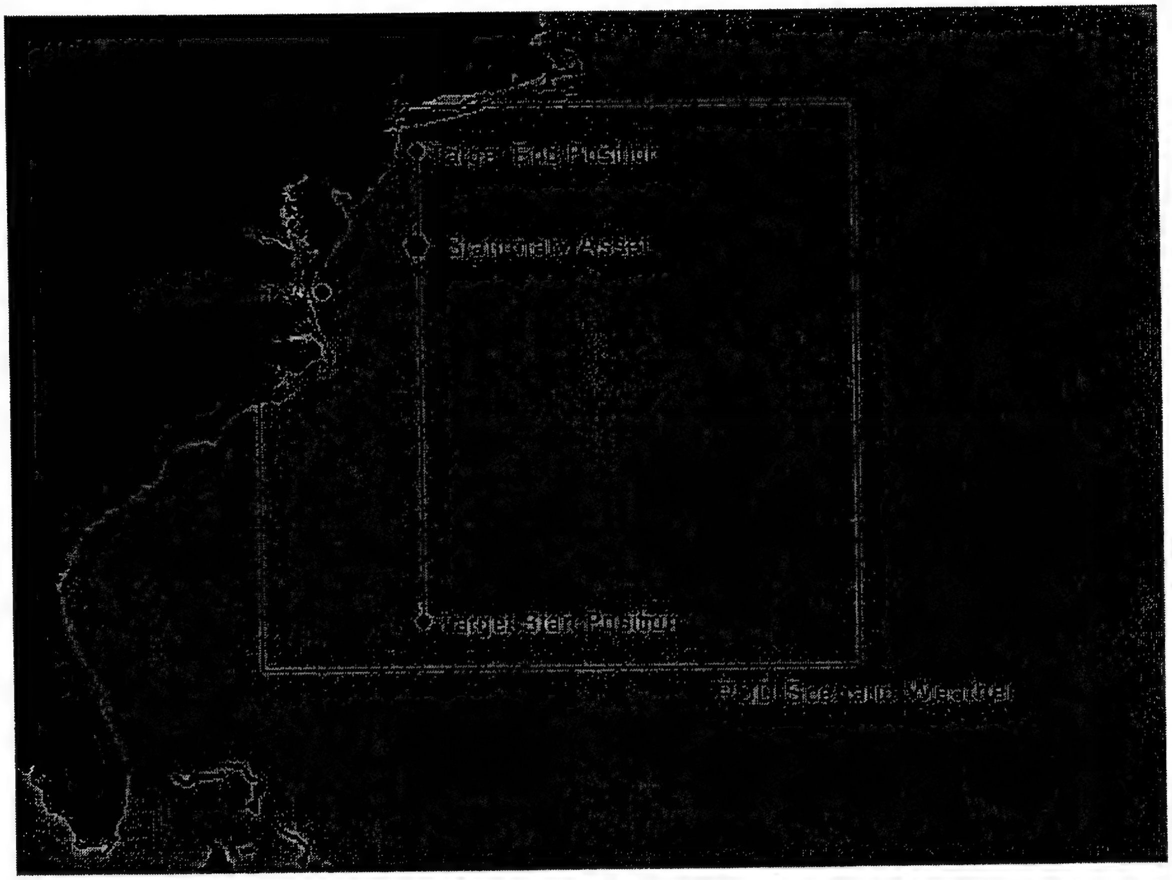

Figure 2: POD vs. Range Scenario

The scenario was set up with the asset directly in the target's path to maximize the number of detections and therefore increase the output data. In addition, each asset is equipped with only one sensor type. Eight different scenarios were run for an entire year with varying target size and sensor height of eye and altitude, as described in **Table 2: POD vs. Range Combinations.**

Table 2: POD vs. Range Combinations

Scenario	Asset	Sensor	Height of Eye / Altitude	Target Size
1	WMEC 270	AN/SPS-73	90	Medium
2	WMEC 270	AN/SPS-73	90	Large
3	WMEC 270	AN/SPS-73	104	Medium
4	WMEC 270	AN/SPS-73	104	Large
5	C 130	APS-137V(4)	1000	Medium
6	C 130	APS-137V(4)	1000	Large
7	C 130	APS-137V(4)	5000	Medium
8	C 130	APS-137V(4)	5000	Large

Once the target is launched, the core determines at what range the asset will make the detection. The distance at which the target is detected is based on several factors, including wave state, precipitation, target size, and asset sensor type and height. For this test, print statements were added to the core detection logic to output the following parameters at time of detection:

- Target size – for this scenario, medium or large
- Time of Day – day or night
- Height of Eye/Altitude – core uses whichever value is greater
- Random Draw – probability of detection
- Distance to Horizon
- Wave Index – Low, Medium, or High
- Precipitation Index – None, Low, Medium, or High
- Lower Limit – lower bracket probability value
- Upper Limit – upper bracket probability value
- Lower Range – range associated with lower bracket probability value
- Upper Range – range associated with upper bracket probability value
- Range – detection range calculated by the core

A resulting database for each of the scenarios was generated. Each row in the database represents a detection attempt by the asset, all relevant variables available to the core, and the detection range (if detection occurred). Each of these rows was compared with the ProbDetectRange table within MarOpsSim to determine if the correct POD vs. Range curve (based on target size, time of day, height of eye, weather [wave index and precipitation index], and probability) was beginning utilized. In addition, the linear interpolating between the probabilities of detections was calculated and verified. This comparison indicates that the proper POD vs. Range curve was being selected and, in turn, the correct range being returned.

4.4.2 Out of Range

In addition to validating the new data structures for the sensor routines, processing surrounding the knowledge of target datum before, during, and after an asset holds a target with its sensor suites has undergone an IV&V.

The following scenario was used to validate the out of range or "undetection" process the core goes through following a target detection. A target traveling due north is launched every 80 hours at a stationary asset located directly in its path. A maneuver in which the target changes course is scheduled for some random latitude between 38.0 and 39.0. The asset is equipped with only one sensor, which has a detection range of 24 nautical miles. The scenario was run for 4,000 hours under fixed weather conditions. **Figure 3: Out of Range Scenario** illustrates the scenario.

Target start position: 37.0, -73.5 Cutter position: 38.0, -73.5 Maneuver Latitude range: 38.0 to 39.0

Target end position; 39.0, - 72.5 Sensor range: 24.0 nmi

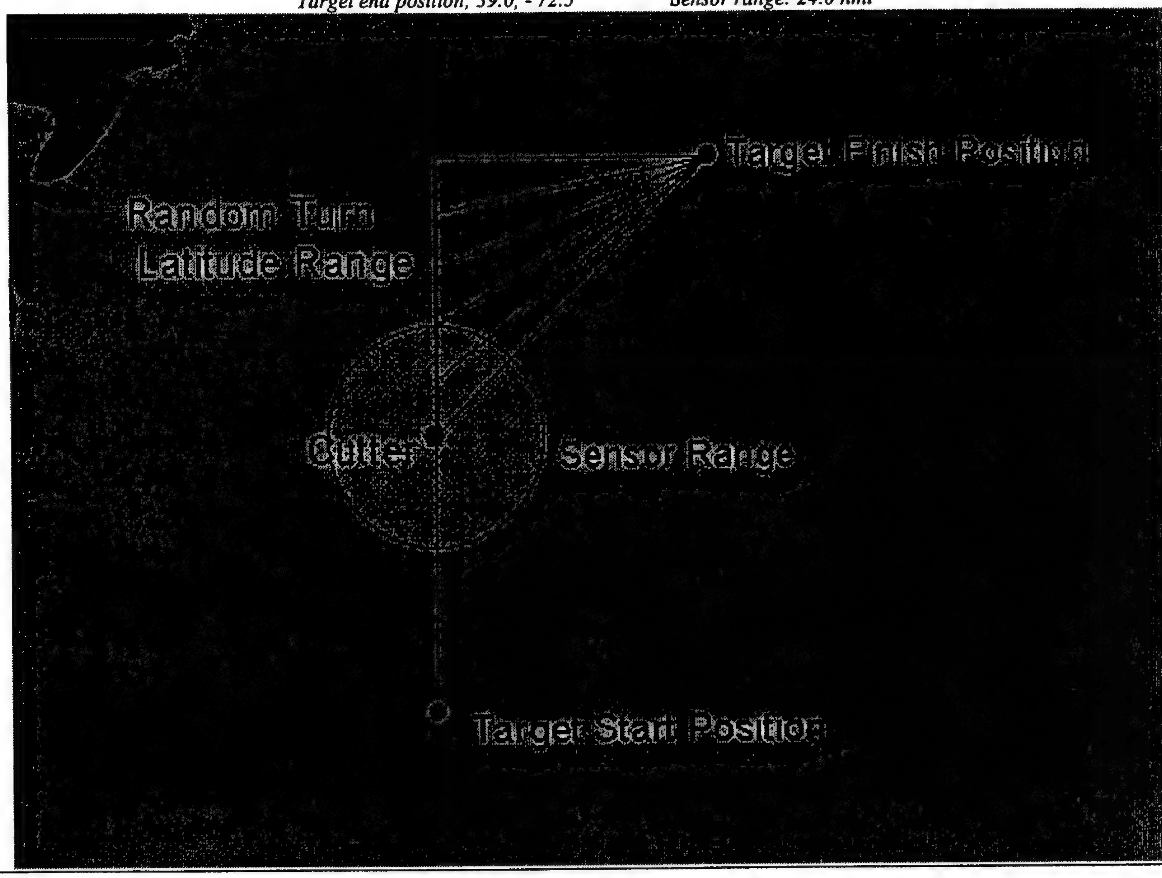

Figure 3: Out of Range Scenario

When a target maneuvers within sensor range, the asset should have accurate knowledge of the target's latitude, longitude, and heading following the maneuver. In contrast, when a target maneuvers outside sensor range, the asset should not accurately report the target's position. The asset will report the target's position based on its last known location. For example, a sensor reporting on a target traveling due north that changes course beyond the sensor's range will continue to report the target's path as due north. The asset will report a change in target's position but this change is based on dead reckoning the target. The asset will not be aware of the change in course.

When the core schedules a detection, it also schedules a complementary out of range event based on the target's course and speed at the time of the detection. Any time the target maneuvers within sensor range subsequent to this, the out of range event is recalculated.

A summary table of results was compiled from the raw results for easier readability. Descriptions of the columns in the summary table are as follows:

- Asset_Target Lat – asset's knowledge of the target's latitude
- Asset_Target Lon– asset's knowledge of the target's longitude
- Asset_Target Hdg – asset's knowledge of the target's heading
- Actual Target Lat – actual target's latitude
- Actual Target Lon – actual target's longitude

- Actual Target Hdg – actual target's heading
- Dist to Target – actual distance to target
- Target Knowledge – knowledge the asset has of the target / 0 indicates no knowledge / 1 indicates detection / 2 indicates classification / 3 indicates identification
- Scheduled ManeuverLat – indicates the latitude where the target will maneuver
- Scheduled ManeuverLon – indicates the longitude where the target will maneuver

By filtering or sorting on the Scheduled ManeverLat and comparing the asset's knowledge of the location of the target to the actual location of the target, it can be determined whether the asset is properly "undetecting" targets. The results show that the core is properly modeling out of range events and accurately maintaining target knowledge based on asset sensor range.

4.4.3 Blind/Invisible

The last sensor behavior to be evaluated is whether the detection-ability state (Normal, Blind, Invisible, or BlindAndInvisible) is properly incorporated in the detection events. This is a very straightforward verification.

The ability to detect and be detected is a function of the sensors a platform is equipped with and the state of a detectability flag. This flag can be set to one of the following:

- Normal – can detect and be detected
- Blind – cannot detect but can be detected
- Invisible – can detect but will not be detected
- BlindAndInvisible – cannot detect and will not be detected

The following scenario was used to validate the detection status of assets and targets. A target traveling due north is launched every 30 hours at a stationary asset located directly in its path. The target is a coastal freighter, which is classified as size "Large." The asset is a WMEC 270 equipped with a sensor with a detection range of 24 nautical miles. **Figure 4: Blind Invisible Scenario** illustrates the scenario.

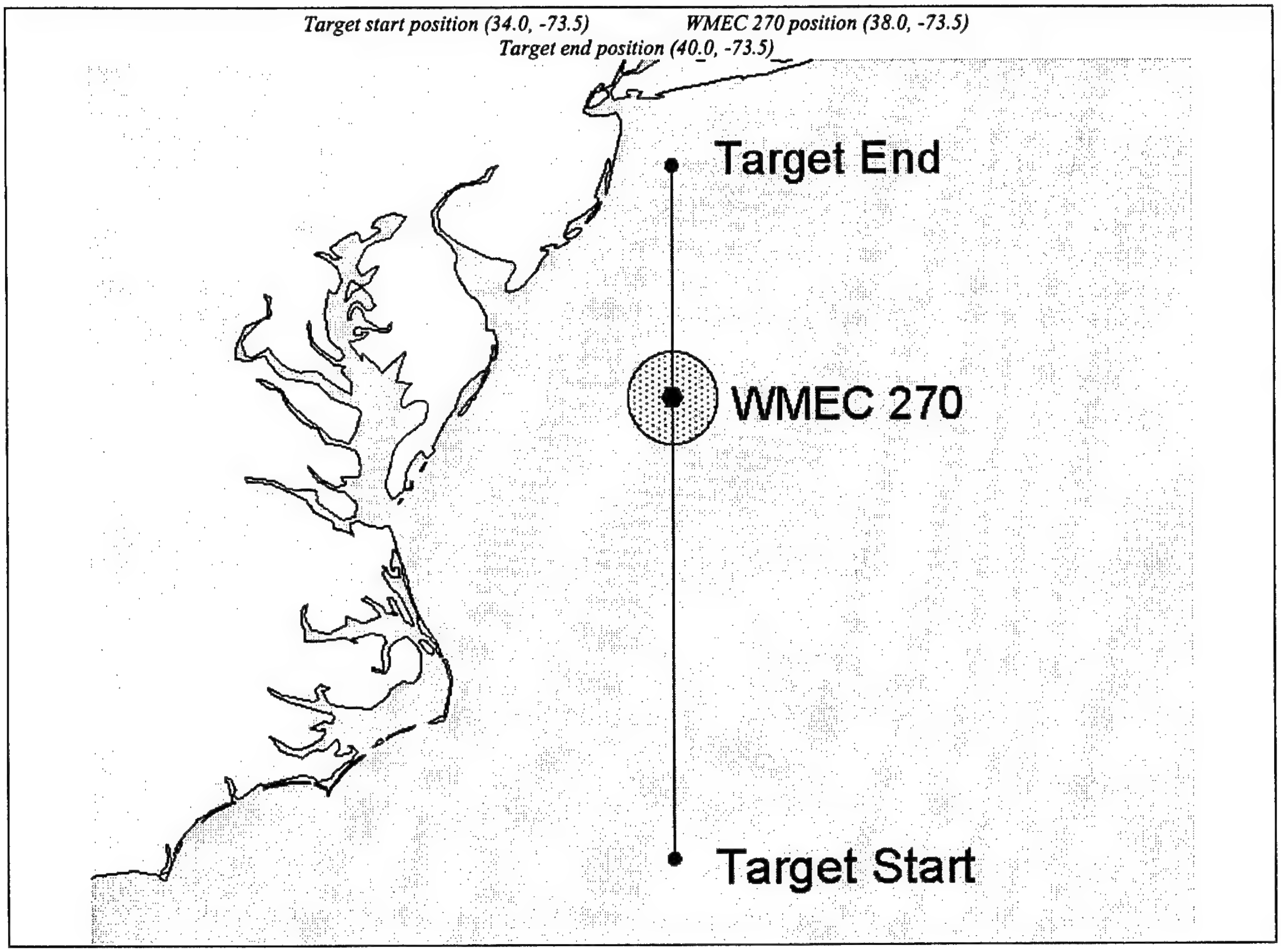

Figure 4: Blind Invisible Scenario

Sixteen scenarios were run for an entire year under varying weather conditions; a total of 100 targets were launched for each scenario. **Table 3: Blind Invisible Scenario Combinations** identifies the combination of values for the detectability flag in each scenario:

Table 3: Blind Invisible Scenario Combinations

Scenario	Asset Detection Status	Target Detection Status
1	BLIND	BLIND
2	BLIND	NORMAL
3	BLIND	INVISIBLE
4	BLIND	BLINDANDINVISIBLE
5	NORMAL	BLIND
6	NORMAL	NORMAL
7	NORMAL	INVISIBLE
8	NORMAL	BLINDANDINVISIBLE
9	INVISIBLE	BLIND
10	INVISIBLE	NORMAL
11	INVISIBLE	INVISIBLE
12	INVISIBLE	BLINDANDINVISIBLE
13	BLINDANDINVISIBLE	BLIND
14	BLINDANDINVISIBLE	NORMAL
15	BLINDANDINVISIBLE	INVISIBLE
16	BLINDANDINVISIBLE	BLINDANDINVISIBLE

A summary database of results was generated. Each row in the database represents a target traveling from the start to finish position, as shown in **Figure 4: Blind Invisible Scenario.** Detection Latitude, Detection Longitude, Detection Time, and Target Detected Status (indicating if the target is detected or not at that time) were all reported into the database. The results show that targets and assets are accurately detecting and being detected based on their detection status.

4.5 Communications

Communications capabilities consist of messages that are sent to platforms that are in range of the sending object. As noted earlier, the scripted behavior dependent on the transmission or receipt of a message needs to be validated. V&V should confirm that the expected behavior does occur, with no unexpected results. Therefore, a scenario was executed for which all the possible outcomes of communications could be expected and was audited to verify that the correct events were in fact occurring. It is to be noted that verifying situations in which communications should not occur is at least as important as verifying situations in which communications should occur. Verifying that communications events occur between assets is the baseline. It is just as important that assets that should not be communicating are not doing so. For example, a communications message sent to an asset that is out of range should never result in a communications event. Since this is a state-dependent situation, the verification scenario should have situations in which communications do occur (they are in range) as well as those in which the same assets do not communicate (they are out of range). Similarly, assets that have incompatible communications devices should never generate communications events regardless of range.

The following scenario was used to validate the core communications. An OPCEN, located in SanJuanPR, sends communications hourly to a WMEC conducting a barrier search, a C130 performing a parallel track search, and a HH65 flying a trisearch off the cutter. The assets and their relative proximity are shown in **Figure 5: Communication Scenario.**

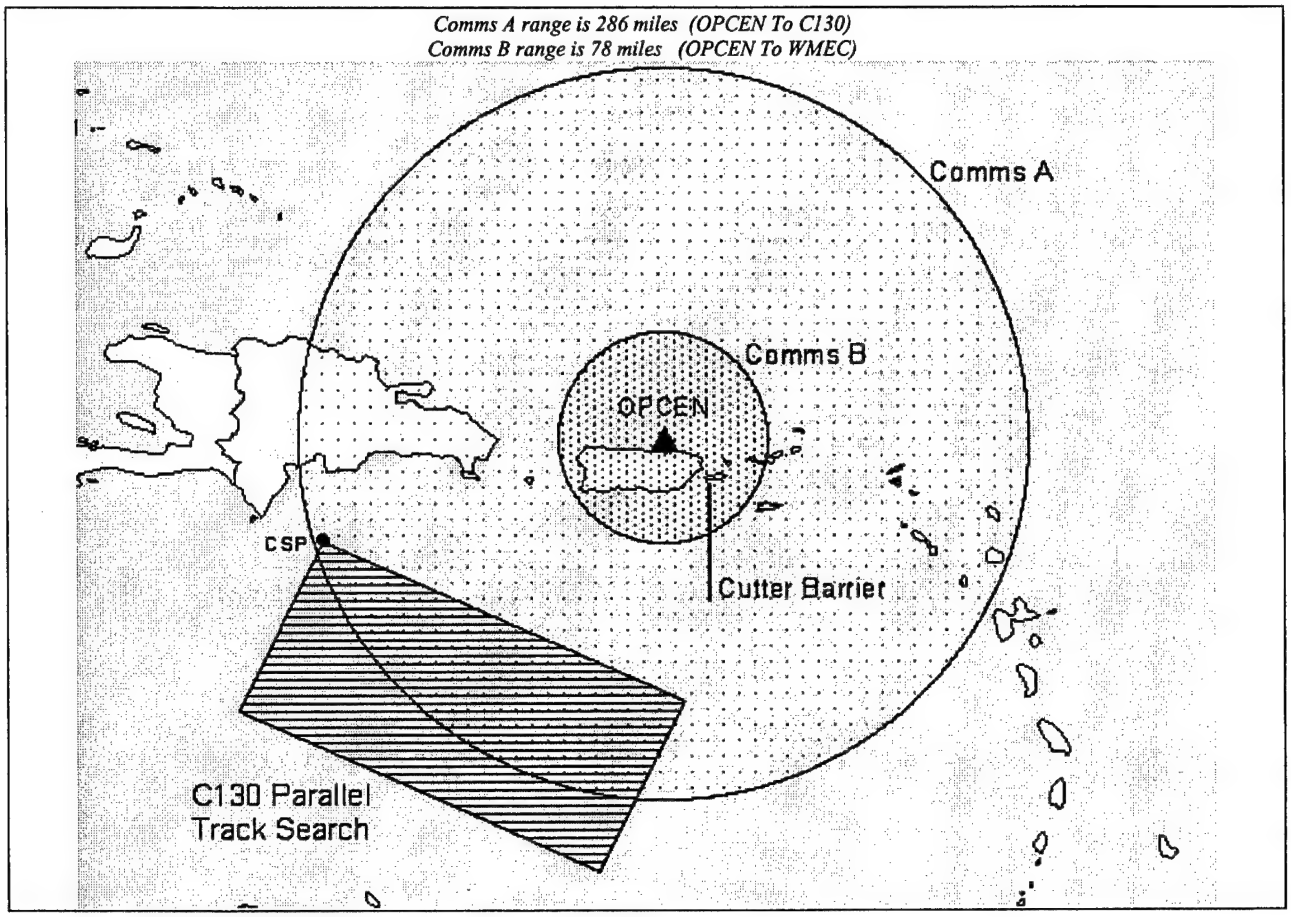

Figure 5: Communication Scenario

The home locations of theses assets are listed in **Table 4: Location of Assets in Communications Scenario.**

Table 4: Location of Assets in Communications Scenario

Asset	Latitude	Longitude	Location
C130	27.93	-83.08	ClearwaterFL
OPCEN	25.72	-79.96	MiamiFL
WMEC	18.39	-65.42	RooseveltRoadsPR
OPCEN	25.72	-79.96	MiamiFL

The communications distances between these assets are shown in **Table 5: Communications Distances Between Scenario Assets.**

Table 5: Communications Distances Between Scenario Assets

	OPCEN	WMEC	C130	HH65
WMEC	78 miles	–	–	41.65 miles
OPCEN	–	78 miles	286 miles	Incompatible
C130	286 miles	–	–	–
HH65	Incompatible	41.65 miles	–	–

The communication paths for the assets are shown in **Figure 6. Communications Paths in Scenario.**

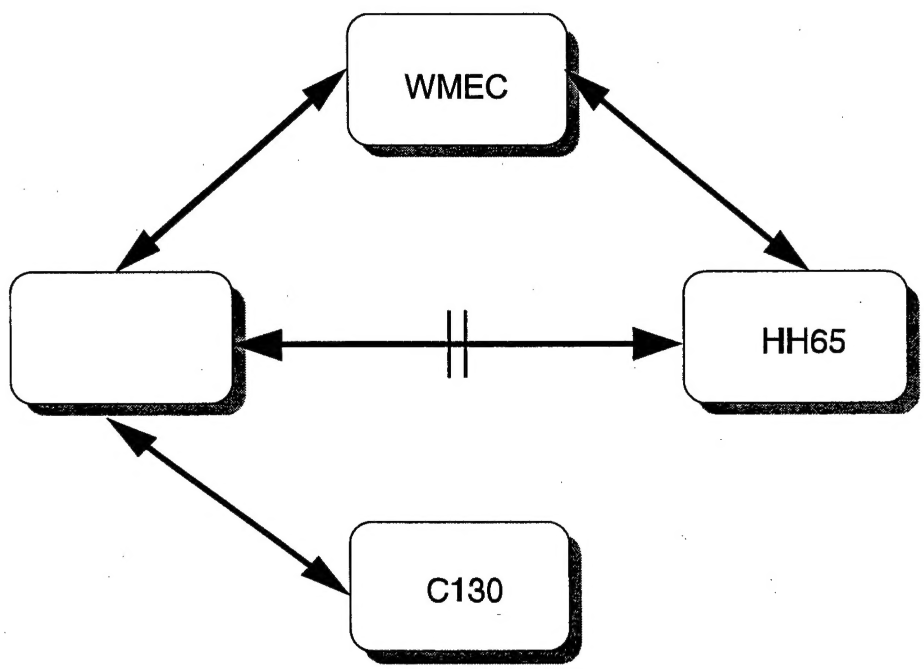

Figure 6. Communications Paths in Scenario

The OPCEN has two communication packages: VHF-FM and HF. For this scenario, the VHF-FM is modeled to use line of sight for its range when sending communications. The other communications package, HF, has a range of 143 miles. The C130 also uses HF to send and receive communications. Messages sent between the OPCEN and C130 can be transmitted a distance of approximately 286 miles. This is calculated by adding the communications range of sender and receiver. The HH65 uses VHF-AM. This is not compatible with either of the communications packages that the OPCEN has. Therefore, communications should never successfully be sent between the OPCEN and HH65. The WMEC is equipped with VHF-FM and VHF-AM. This combination of communications packages allows messages to be sent between OPCEN and WMEC and between WMEC and HH65. Both VHF-FM and VHF-AM use line of sight to determine the communications range. Line of sight is calculated using either height of eye or altitude, whichever is greater.

To determine line of sight, the core uses the following calculation:

$$LOS = 1.1444\sqrt{\max(H, A)}$$

where H = height of eye and A = altitude

Using the WMEC to OPCEN communications as an example, line of sight is calculated as follows:

WMEC height of eye – 81.4 feet

OPCEN height of eye – 3500 feet

WMEC line of sight = $1.144\ \sqrt{81.4} = 10.32$ miles.

OPCEN line of sight = $1.144\ \sqrt{3500}\ = 67.68$ miles.

Distance communications can be sent between OPCEN and WMEC = 10.32 + 67.68 = 78 miles.

Altitude is used to determine line of sight when its value is greater than height of eye, as in the following example (using HH65 altitude as 750 feet):

HH65 line of sight = $1.144 \cdot \sqrt{750} = 31.33$ miles

When the WMEC receives communications from the OPCEN, it will forward the message to the helicopter if the HH65 is flying. When the HH65 is on the cutter, messages are not forwarded. Upon receipt of communications from the WMEC, the helicopter will forward the message back to the cutter. When the cutter receives a message from the helicopter, it will send the message to the OPCEN. The scenario is such that the OPCEN will never successfully send communications to the helicopter, as the communication packages are not compatible. Upon receipt of communications from the OPCEN, the C130 will send the message back to the OPCEN.

The scenario was run for 1,000 hours, and a database of summary results was generated. Each row in the database represents an attempt to send a communications message. Sender, receiver, sender and receiver latitudes and longitudes, distance between sender and receiver, and communications success/failure are included for each message. The results show that the core is properly modeling communications between the assets. Messages are being sent when in range and failing to be sent when out of range. Communications were also shown to fail when incompatible communications packages exist between sender and recipient.

Four tables of raw results and a summary table were generated. Each row in the summary table represents an attempt to send a communications message. Sender ID, receiver ID, sender and receiver latitudes and longitudes, and distance between sender and receiver are included for each message sent. A Communications Flag was also reported. This is the result of the *SendMessage* script command as described in Section 5.2.2.7.2, page 42, of the CSDD. The value of the Communications Flag indicates the number of successful transmission paths; therefore, a "1" or "2" would represent that a communications message was sent, and a "0" would designate a communications failure. In addition, distance (d) between sender and receiver was calculated using the same formula as the core:

$$\Delta Lat = 60(Lat_2 - Lat_1)$$

if sender or receiver longitude is negative, add 180

$$\Delta Lon = 60.0 \cdot (Lon_2 - Lon_1) \cdot \cos(0.5(Lat_2 - Lat_1))$$

$$d = \sqrt{\Delta Lat^2 + \Delta Lon^2}$$

The summary results table shows times when messages are successfully sent at distances greater than the expected range of the communications packages. This occurs for both the WMEC to OPCEN and C130 to OPCEN. On reviewing these occurrences, communications were successfully transmitted

because both sender and recipient where in port/air station and were able to communicate through a direct (land line) means. When an asset is in port/air station, the model assumes that it can communicate with other assets in port/air station or permanently land based, such as an OPCEN.

The summary results table shows that the core is properly modeling communications between the assets. Messages are being sent when in range and failing to be sent when out of range. Communications were also shown to fail when incompatible communications packages exist between sender and recipient.

4.6 Weather

There has been no change to the weather model itself. The changes of POD/Range data structures are weather dependent and therefore incorporated into the sensor revalidation.

5 Conclusions

Based on the analysis and assessment made so far, it can be concluded that MarOpsSim is a modeling tool that can be used to represent the characteristics and behavior of the Coast Guard against the demands and in the operating environments described in the MSMP. Furthermore, MarOpsSim produces output that can be summarized and analyzed to consistently generate Coast Guard system performance measures also described in the MSMP.

6 References

[1] MicroSystems Integration, Inc. Maritime Operations Simulation (MarOpsSim) Consolidated Software Design Document (December 2000).

[2] Deepwater Project Office. United States Coast Guard Deepwater Capability Replacement Analysis: Modeling and Simulation Master Plan (MSMP), Change 9 (March 2001).

[3] Buss, A. & T. Halwachs. Core Validation for Maritime Operations Simulation (MarOpsSim), Technical Report NPS-OR-00-093 (November 1999).

DISTRIBUTION LIST

1. Research Office (Code 09) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 013) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 1
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Richard Mastowski (Editorial Assistant).. 2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Professor Arnold H. Buss (Code OR/Bu).. 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Professor Thomas E. Halwachs (Code OR/Ha).. 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. MicroSystems Integration, Inc. 2
   158 South Broad Street, Suite C
   Pawcatuck, CT 06379-1925

8. MicroSystems Integration, Inc. 1
   158 South Broad Street, Suite C
   Pawcatuck, CT 06379-1925
   Attn: LCDR Phil Muir

9. U.S. Coast Guard Research & Development Center 3
   Maritime Operations Technology Division
   1082 Shennecossett Rd.
   Groton, CT 06340
   Attn: Kevin Downer